Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 50 (2015) 81 – 86

Procedia Computer Science

2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

A Secure and Light Weight Authentication Service in Hadoop using One Time Pad

Mrudula Sarvabhatla[a], Chandra Mouli Reddy M[b], Chandra Sekhar Vorugunti[c]

[a]*N.B.K.R Institute of Science and Technology, Nellore, AP, 524413, India*

[b]*Veltech University, Avadi, Chennai,TN,-600 062, India*

[c]*Dhirubhai Ambani Institute of Information and Communication Technology, Gandhi Nagar, Gujarat, 382007,India*

Abstract

To store and process sensitive data related to credit card, healthcare, financial, government which constitutes the big data, Hadoop environment is supplemented with HDFS i.e Hadoop Distributed File System, which is a cloud based, parallel processing, open framework. As the HDFS stores and process critical sensitive data, there is a great need for secure authentication service to authenticate and authorize the user connecting to HDFS. In 2014, Nivethitha et al has proposed first of its kind of authentication service for Hadoop using one time pad. Even though it is a novel attempt, in this paper, we will illustrate that, Nivethitha et al scheme is vulnerable to offline password guessing attack and on success of it, an attacker can perform all major attacks on HDFS. As a part of our contribution, we will device a new authentication service for hadoop framework which is light weight and resists all major attacks compared to Nivethitha et al. authentication service while maintaining the merits of their authentication service.

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Keywords: Hadoop Framework;One Time Pad; Authentication Service; Big Data

1. Introduction

The rapid augmentation of online transaction-oriented data, flooding of unstructured data from online social media, exponential evolution of real time data from sensor networks and machine-to-machine communications results in variety of voluminous data collectively called Big data. Big data cannot be stored or processed through conventional storage or processing systems. To cater the enterprise data storage and processing needs, Apache developed Hadoop project which is a java based, open-source framework for distributed storage and processing of

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

doi:10.1016/j.procs.2015.04.064

enterprise data. In Hadoop framework, all the file storage activities are handled by HDFS i.e Hadoop distributed file system, which can store peta bytes of sensitive enterprise data.

Preliminary security analysis [1,2,3,4] on Hadoop reveals that, to achieve strong authentication and authorization mechanism in Hadoop, there is acute need of secure and cost effective authentication service.

In literature [1,2,3,4] there are four kinds of system architecture has been proposed for authentication service. Single server authentication service, simple multi-server model, gateway augmented multi-server model and two server model. Each architecture is having one or more pitfalls associated with it. Among them two server model is found to be effective [1,2,3].

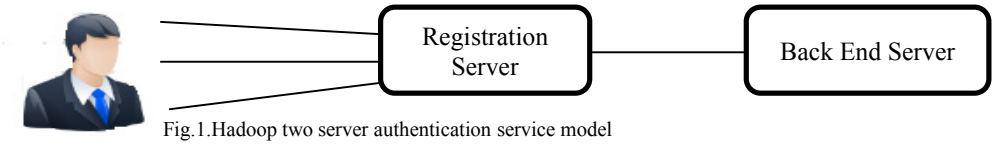


Fig.1.Hadoop two server authentication service model

Very few researchers have proposed authentication service for Hadoop framework based on the two server model. Nivethitha et al [1] has proposed a first of its kind of authentication service for Hadoop using One Time Pad. A user password is encrypted with the generated OTP in registration server and transmitted securely to the backend server. Nivethitha et al claimed that their Hadoop authentication service is secure and attack resistant. Sadasivam et al [2] has proposed an authentication service for Hadoop in cloud environment based on two server model, grounded on properties of a triangle, in which the user password separated in two back end servers to resists the attacks.

In these next segments, we briefly review on authentication service proposed by Nivethitha et al [1] in section 2. In section 3, we do cryptanalysis on Nivethitha et al scheme. In section 4, we propose our improved authentication service for Hadoop. In section 5, we scrutinize the security strengths of our proposed framework. In section 6, we do a computation and communication cost analysis on our proposed framework and Nivethitha et al [1] authentication service. Section 7 provides the conclusion of our paper.

2. Brief review of Nivethitha et al authentication service

In this segment, we inspect the Nivethitha et al. authentication service based on two server model depicted in fig 1.

Registration Stage:

User U_i **Registration Server (RS)** **Backend Server (BS)**

Select ID_i, PW_i
Submit $\{ID_i, PW_i\}$ to R.S
$\{ID_i, PW_i\}$ → $\{ID_i, PW_i\}$ →

Generate One Time Pad i.e OTP.
$C1 = E_{OTP}(PW_i)$, $C2 = C1 \bmod 26$.
Store <C2> in database

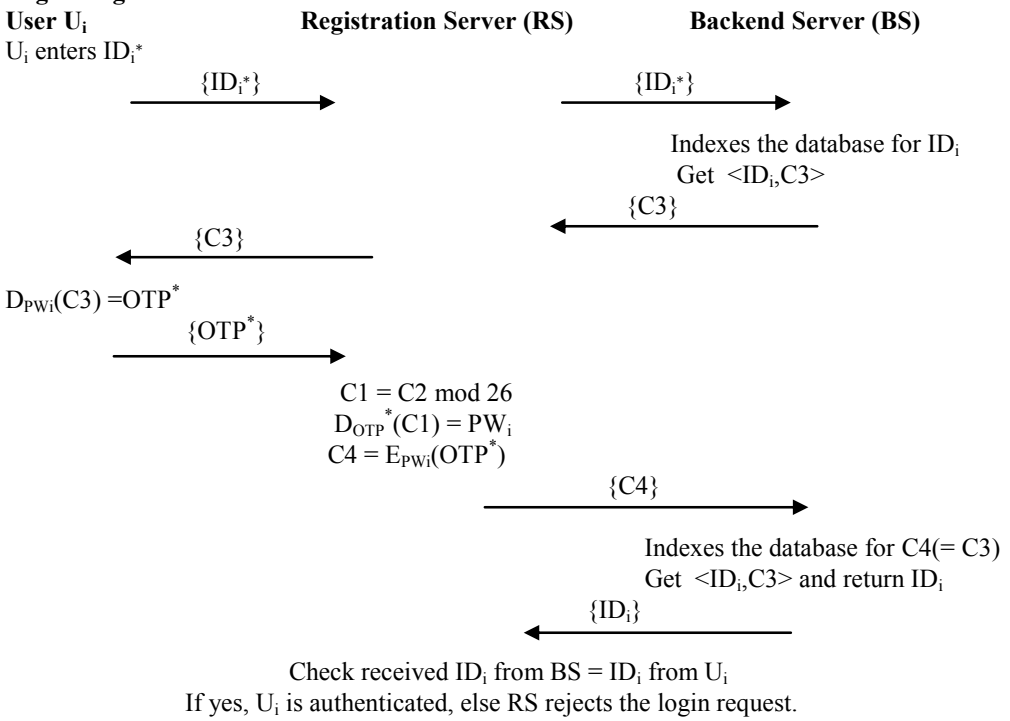
$\{OTP\}$ →

$C3 = E_{PWi}(OTP)$
Store $<ID_i, C3>$ in database

This phase is invoked whenever a user U_i registers with Hadoop framework via registration server (R.S) for the first time.
(R1). U_i opts his identity ID_i, password PW_i and submits the registration request $\{ID_i, PW_i\}$ to RS, R.S forwards the same to B.S.

(R2) On receiving the registration request, R.S generates one time pad key i.e OTP and encrypts the user password PW_i with the OTP generated i.e ciphertext $C1 = E_{OTP}(PW_i)$ and generates cipher text C2 by applying mod on C1. i.e$C2 = C1 \bmod 26$. R.S stores cipher text <C2>in its database securely. R.S forwards OTP to B.S.
(R3) B.S computes cipher text C3 by encrypting the OTP with the user password PW_i i.e= $E_{PWi}(OTP)$ and stores $\{ID_i,C3\}$securely in its database.

Login Stage:



The user U_i wish to login into Hadoop environment needs to perform the following computations:
(L1) U_i submits $\{ID_i^*\}$ to registration server R.S. R.S forwards the same i.e$\{ID_i^*\}$to B.S.
(L2) On receiving the login message i.e$\{ID_i^*\}$ from R.S, B.S indexes its data base for <ID_i^*,C3> which it stored during login stage. B.S returns {C3} to R.S. On receiving the reply from B.S, R.S forwards the same i.e {C3} to U_i.
(L3) On receiving the reply message {C3} from R.S, U_i decrypts C3 using its password PW_i i.e $D_{PWi}(C3)$ to get OTP^*. U_i forwards the same to R.S i.e$\{OTP^*\}$. On receiving $\{OTP^*\}$, R.S performs reversal operation on C2 to get C1. R.S performs decryption of C3 using the OTP^* sent by U_i i.e $D_{OTP}^*(C1) = PW_i$.
(L4) On getting U_i password i.e PW_i, R.S computes $C4 = E_{PWi}(OTP^*)$ and forwards {C4} to B.S.
(L5) B.S indexes for C4 (=C3) and retrieves ID_i from <ID_i,C3> combination, stored during registration phase and forwards the same to R.S. On getting ID_i from B.S, R.S cross checks, whether the received ID_i from BS equals ID_i^* received from U_i. If both are equal, U_i is authenticated, else RS rejects the login request.
(L6) Once U_i is authenticated and logs out after completing his action, R.S generates a new OTP i.e OTP^{new} and computes $C1^{new} = E_{OTP}^{new}(PW_i)$, $C2^{new} = C1 \bmod 26$and updates its database value < C2> to <$C2^{new}$>.

3. Cryptanalysis of Nivethitha et al framework

In this segment, we scrutinize the security strengths of Nivethitha et al [1] scheme, for that we will follow the similar threat model practiced by [3]:

1. An opponent can monitor or eavesdrop or intercept all the messages exchanged between U_i and R.S, as these messages are exchanged over public internet. The messages exchanged between registration server R.S and back

end server B.S are not accessible to the public as these messages are assumed to be exchanged in private network internal to Hadoop environment. i.e the attacker 'E' is having access to $\{ID_i^*\}$, $\{C3\}$,$\{OTP^*\}$ as these are the messages exchanged between U_i and R.S during login stage.

3.1 Fails to Resists Offline Password Guessing Attack

An attacker or opponent 'E', as discussed, can intercept the messages $\{ID_i\}$, $\{C3\}$,$\{OTP\}$. On achieving $C3 = E_{PWi}(OTP)$, 'E' can accomplish the subsequent steps to intercept the U_i password PW_i:
Step1: Guess a password PW_g^* from a uniformly distributed dictionary.
Step 2: Compute $D_{PWi^*}(C3) = OTP^*$, and verify whether OTP^* is equal to OTP intercepted. If yes, the U_i original password is PW_g^* else 'E' re run the execution of step1.
Once the attacker 'E' guess the correct password, on having $\{ID_i, PW_i\}$, 'E' can login into the system by impersonating U_i. 'E'can also successfully decrypt the reply message $\{C3\}$ sent by R.S to U_i during login phase.

4. Our Enhanced Scheme

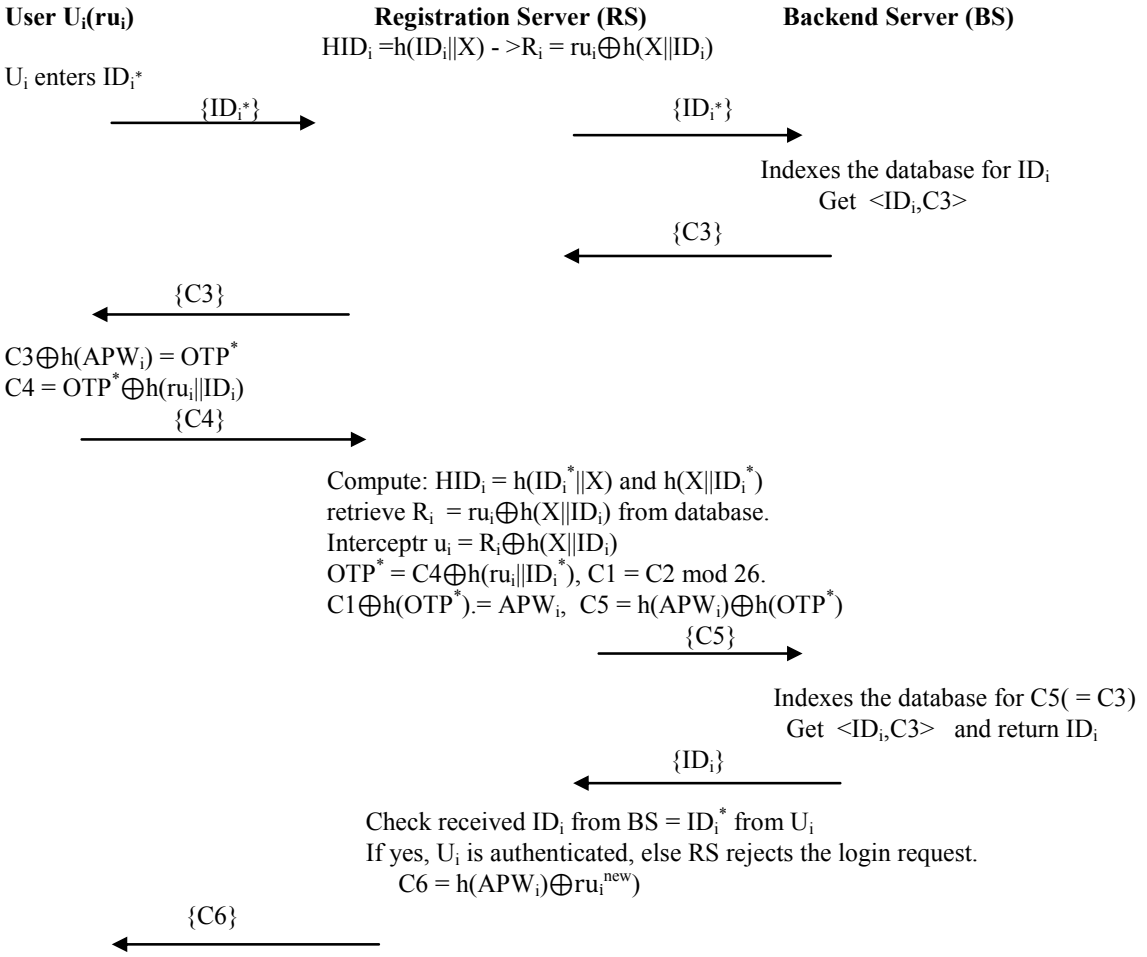
In this segment, we portray our proposed scheme. The main rationale of our enhanced scheme is to resist the security flaws set up in Nivethitha et al [1] scheme.

Registration Stage:

User U_i	**Registration Server (RS)**	**Backend Server (BS)**
Select ID_i, PW_i,a random number a_i $APW_i = h(a_i \oplus PW_i)$		
$\{ID_i, APW_i\}$ →		
	Generate One Time Pad i.e OTP. $CID_i = ID_i \oplus h(OTP)$ $C1 = APW_i \oplus h(OTP)$. C2 = C1 mod 26 store <C2> in database	
	$\{CID_i, APW_i, OTP\}$ →	
		$C3 = h(APW_i) \oplus OTP$ $ID_i = CID_i \oplus h(OTP)$ stores<ID_i,C3> securely in its database.
	Generates a random number $\{ru_i\}$	
$\{ru_i\}$ ←		
		Compute $HID_i = h(ID_i \| X)$ stores $ru_i \oplus h(X \| ID_i)$ against HID_i in R.S database.

(R1). U_i opts his identity ID_i, password PW_i, an arbitrary number a_i and submits the registration request $\{ID_i, APW_i = h(a_i \oplus PW_i)\}$ to RS.
(R2) On receiving the registration request, R.S generates one time pad key i.e OTP and computes $CID_i = ID_i \oplus h(OTP)$, $C1 = APW_i \oplus h(OTP)$, C2 = C1 mod 26.R.S stores <C2> in its database. R.S forwards $\{CID_i, APW_i, OTP\}$ to the back end server (B.S). On receiving the message $\{CID_i, APW_i, OTP\}$, B.S computes $C3 = h(APW_i) \oplus OTP$ and $ID_i = CID_i \oplus h(OTP)$ and stores <ID_i,C3> securely in its database.
(R3) R.S generates a random number ru_i and stores in it database and forwards the same to U_i.

Login Stage:



Both U_i and R.S updates their entries with new random number i.e ru_i^{new}.

The user U_i need to login into Hadoop environment, U_i performs the following computations:
(L1) U_i submits $\{ID_i^*\}$ to registration server R.S. R.S forwards the same i.e $\{ID_i^*\}$ to B.S. On receiving the login message i.e $\{ID_i^*\}$ from R.S, B.S indexes its data base for $<ID_i, C3>$ which it stored during login stage. B.S returns $\{C3\}$ to R.S. On receiving the reply from B.S, R.S forwards the same i.e $\{C3\}$ to U_i.
(L2) On receiving the reply message $\{C3\}$ from R.S, U_i computes $C3 \oplus h(APW_i) = OTP^*$, $C4 = OTP^* \oplus h(ru_i||ID_i)$ where ru_i is the random number assigned by R.S to U_i during registration stage.U_i forwards $\{C4\}$to R.S.
(L3) On receiving $\{C4\}$, R.S computes $HID_i = h(ID_i^*||X)$, $h(X||ID_i^*)$ and retrieves ru_i from $R_i = ru_i \oplus h(X||ID_i)$ which is stored in its database. On getting ru_i, R.S computes $OTP^* = C4 \oplus h(ru_i||ID_i^*)$, performs reversal operation on C2 to get C1.
(L4) On computing C1, R.S proceeds to get $APW_i = C1 \oplus h(OTP^*)$ and computes $C5 = h(APW_i) \oplus h(OTP^*)$. R.S forwards $\{C5\}$ to B.S. B.S indexes for C5 (=C3) and retrieves ID_i from $<ID_i, C3>$ combination, which is stored during registration phase and forwards the same to R.S.
(L5) On getting ID_i from B.S, R.S cross checks, whether the received ID_i from BS equals ID_i^* received from U_i.
If both are equal, U_i is authenticated, else RS rejects the login request. Once U_i is authenticated and logs out of Hadoop, after completing his action, R.S generates a new random number ru_i^{new} and computes $C6 = h(APW_i) \oplus ru_i^{new}$). R.S forwards $\{C6\}$ to U_i and updates it database entry for U_i with ru_i^{new}.

5. Security Analysis of Improved Scheme

5.1 Resistance to Offline Password Guessing Attack

An attacker 'E', can intercept the messages $\{ID_i^*\}$, {C3} and {C4} which are exchanged between U_i and R.S over public internet. On achieving ID_i^*, $C3 = h(APW_i) \oplus OTP = h(h(a_i \oplus PW_i)) \oplus OTP$, $C4 = OTP^* \oplus h(ru_i || ID_i)$. To perform password guessing attack on C3, the attacker must know a_i, ru_i. But, a_i, ru_i are random numbers of length 128 bits. Hence, it is computationally impossible for an attacker to correctly guess a_i or ru_i. Therefore, our scheme resists password guessing attack.

5.2 Counter to Replay Attack

An attacker 'E' wish to replay a login message from U_i, i.e{C4} to R.S, as discussed in (L6), after every successful login of U_i, the random number ru_i gets changed. R.S may start processing a replay message from the attacker 'E' and compute C5 out of replayed C4 message and forward {C5} to B.S. On receiving {C5}, B.S searches in its database for {C5}, definitely B.S doesn't find an entry for C5 (C3) in it database. Suppose if it finds an entry, definitely it is not for ID_i. (As the replayed message C5, and C3 stored in back end server are computed using different random numbers). Hence, B.S doesn't responds with a valid ID_i. Therefore, R.S will reject the login request by 'E'. Hence, we can confirm that our scheme is resistant to replay and impersonation attacks.

6. Cost and Security analysis

The cost comparison between the proposed user authentication service and Nivethitha et al [1] authentication service, illustrates that our authentication service requires only 15 hash operations compared to 50 hash operations by Nivethitha et al service. As discussed above, our service is very secure and resists all major cryptographic attacks with less computation cost compared to Nivethitha et al scheme. (in which the heavy weight cryptographic operations like encryption and decryption form the major computation force).

7. Conclusion

In this manuscript, we have analyzed Nivethitha et al HDFS authentication scheme using one time pad (OTP). We have illustrated that their scheme is vulnerable to password guessing attack and on success of it, the attacker can perform all major attacks. To fix these issues, we have proposed a light weight OTP based authentication service for hadoop environment. We have analyzed our scheme in terms of both security front and cost front and illustrated that our authentication service resists all major cryptographic attacks. Hence, we can confirm that, our authentication service is light weight and robust and can be implemented in practical scenarios.

References

1. Nivethitha S, Gangaa A, Shankar S, Authentication Service in Hadoop Using one Time Pad, Indian Journal of Science & Technology ,vol 7, pp 56-62, Apr 2014.
2. Sadasivam G.S, Kumari K.A, Rubika S, A Novel Authentication Service for Hadoop in Cloud Environment, IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp 1-6, oct 2012,India.
3. Zhou Quan, Deqin Xiao, hunming Tang, ChunmingRong,TSHC: Trusted Scheme for Hadoop Cluster,Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT),9-11 Sept. 2013.
4. Rahul P.K, GireeshKumar T, A Novel Authentication Framework for Hadoop, International conference on Intelligence and Evolutionary Algorithms in Engineering Systems (ICAEES 2014) , Vol 324, 2015, pp 333-340 .